

BioPARKIN Release 1.2

User Manual

Computational Systems Biology Group
csb@zib.de

31st October 2012

Zuse Institute Berlin, Takustr. 7, 14195 Berlin

Abstract

BioPARKIN is a software package for simulation and parameter identification in ordinary differential equation models with special emphasis on systems biology. It combines state-of-the-art numerical routines with a graphical user interface that allows intuitive handling of models including species, parameters, and mechanisms. The use of the SBML file format ensures compatibility with other tools available for systems biology. BioPARKIN is available for Windows, Mac OS X, and LINUX. This document provides a user manual for release 1.2.x.

Contents

1	Introduction and Background	2
2	Formats	3
2.1	SBML Compatibility	3
2.2	Input and Output of Experimental or Simulated Data	4
3	Graphical User Interface - Overview	5
3.1	Model Overview	5
3.2	Workbench	5
3.3	Results Window	7
4	Graphical User Interface - Usage	7
4.1	Model Overview: Working with a Model	7
4.2	Workbench: Settings	9
4.2.1	General Settings	9
4.2.2	Time Settings	11
4.2.3	Plot Settings	11

4.2.4	Expert Settings	11
4.2.5	Simulation	12
4.3	Workbench: Species	13
4.4	Workbench: Parameters	13
4.5	Workbench: Parameter Sets	13
4.6	Workbench: Data Browser	16
4.7	Workbench: Sensitivity	16
4.8	Workbench: Fit	20
4.8.1	NLSCON	20
4.8.2	PARKIN	21
4.9	Results Window	21
4.9.1	Data Plots	21
4.9.2	Data Tables	24
References		24
A Problem Description		27
A.1	Large Kinetic Networks in Systems Biology	27
A.2	Multiple Experiments	27
A.3	Breakpoint Handling	27
A.4	Sensitivity Matrix	28
A.5	Parameter Identification	28
A.6	Parameter Constraints	30

1 Introduction and Background

Modelling, parameter identification, and simulation play an important role in systems biology. In recent years, various software packages have been established for scientific use in both licencing types, open source as well as commercial. Many of these codes are based on inefficient and mathematically outdated algorithms. By introducing the package BioPARKIN we want to improve this situation significantly. The development of the software BioPARKIN involves long standing mathematical ideas that, however, have not yet entered the field of systems biology, as well as new ideas and tools that are particularly important for the analysis of the dynamics of biological networks.

Systems biology aims at describing biological processes by mathematical models, which permit biologically sound quatitative predictions. To arrive at statements of this kind, the time courses of biological processes are modeled by differential equations that describe the concentrations or amounts of the involved substances over time. For the time being, we consider only systems of ordinary differential equations (ODEs). Delay differential equations (DDEs) or differential algebraic systems (DAEs) are not supported until future releases of BioPARKIN. In BioPARKIN, the ODE systems are solved numerically with LIMEX [5], a linearly implicit Euler method with extrapolation, which is especially suited for stiff systems.

From a mathematical point of view, the main difficulty is not to simulate such systems, i.e. to solve the differential equations, but to determine the unknown parameters in such a way that the simulation results agree with experimental measurement values. This is an inverse problem which can be formulated as a nonlinear least squares problem. In biological models, there are usually dependencies between parameters, which leads to rank-deficient problems. For the solution of such problems, efficient and

reliable numerical algorithms (Gauss-Newton methods) based on affine invariant Newton methods have been developed over the past decades [1]. An error-oriented algorithm with adaptive trust region and rank strategies has been implemented in the code NLSCON (Nonlinear Least Squares with CONstraints) [1]. A precursor version of NLSCON had already been part of the software package PARKIN [12, 4, 11], a single shooting method for parameter identification in large chemical reaction kinetic networks. The two methods differ in the damping strategy of the Newton steps. NLSCON allows the user to control the algorithmic setting, whereas PARKIN works much more in an automated way, which might not always be suitable for the problem at hand. Both methods have been integrated into BioPARKIN and form the numerical core of the software. Thus, BioPARKIN can be considered as a successor of PARKIN especially designed for applications in systems biology. In particular, we aimed at a comprehensive and flexible implementation by object-oriented design. To summarise, BioPARKIN contains a stand-alone library written in C++ which deals most efficiently with the following numerical tasks:

- fast expression evaluation,
- solver LIMEX for stiff differential-algebraic systems,
- Gauß-Newton methods NLSCON and PARKIN for parameter identification.

The current release of BioPARKIN is not able to detect if a network of reactions is too large, i.e. there are redundant reaction links that could be eliminated without any change in the results. However, BioPARKIN is designed to indicate and to correctly deal with the situation where the parameter space contains a non-trivial subspace of solutions to the inverse problem at hand rather than only exact one solution, even if actually accurate measured data is available. In this case, the parameter identification can be improved, and the determined parameters are no longer open to any interpretation.

If a network is too small, i.e. important compartments, species, and/or reactions are missing in order to explain the given data, this eventually shows up in poor convergence of the Gauß-Newton method during parameter identification.

BioPARKIN combines the state-of-the-art numerical routines with a graphical user interface that allows intuitive handling of models including species, parameters, and mechanisms. How to navigate through this interface will be the main topic of this manual. Section 2 explains the compatibility of BioPARKIN with SBML and other software. Section 3 gives an overview of the GUI, whereas model handling, simulation and parameter identification are described in detail in Section 4. For the well disposed user, the mathematical background is explained in Appendix A.

2 Formats

BioPARKIN uses SBML (Systems Biology Markup Language)¹ to handle model files, and a tab-separated value format to load and store experimental data and simulation results. The two formats will be explained in this section.

2.1 SBML Compatibility

During the last decade, it became clear that well-defined standards would benefit the systems biology community. Independent from each other, several projects arose to create a standard for modelling biological systems and sharing them with the community.

¹M. Hucka et al.: The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models, 2003. www.sbml.org

Timepoint [unit]	ID1 [unit]	SD	ID2 [unit]	SD	Arbitrary Column
0	0	1	0	1	...
3.33	0.4	1	0.09	1	...
...	...	...	...	...	...

Table 1: Structure of a BioPARKIN data file. Neighbouring cells are separated by a tab character.

While other formats are still maintained and progressed, SBML is the de-facto standard—especially in simulation-centric workgroups and projects [14]. The SBML standard is supported by a lively community of researchers and software engineers. On one hand, there is the official software library, *libSBML*, to access SBML files and interact with them. This library is maintained by a central core of the community and is vital to almost all other software projects that work with SBML. On the other hand, several dozen tools exist to work with SBML models. Most of these tools are developed independently and serve a specialised niche use case.

Currently, BioPARKIN supports a subset of SBML Level 2 Version 4 [8]. SBML’s event definitions, for example, support more types of events than BioPARKIN currently can handle. Situations that the software cannot handle are usually reported by errors or warnings within the user interface or the log file.

Regardless of which standard is used, the researcher has to know the syntax and terminology of the format. SBML defines important entities like Species, Reactions, Compartments, Kinetic Laws, Rate Rules, Assignment Rules, Events, and others. BioPARKIN has adopted these terms and we will come across them later in this manual.

On the upside, almost all SBML-compatible tools share the same terminology. This makes it easier to use new tools and it also facilitates the communication between researchers that use different tools.

BioPARKIN ensures compatibility with other tools available for systems biology, e.g. *COPASI* (*COmplex PATHway Simulator*) [7], which has a large variety of features for simulation and analysis of biological models. Another useful general-purpose SBML tools is *CellDesigner*.

CellDesigner is one of the most-widely used SBML tools to model and display SBML models [6]. It can also be used to simulate models using the *SOSlib* library [10]. It adheres to the *SBGN* standard that defines the visual representation of SBML entities[9]. Although it supports the simulation of models it cannot compete with the numerical features that BioPARKIN offers. It is more suited to get a quick impression of the model’s simulation results. In this way, *CellDesigner* is an ideal complement to BioPARKIN. Whenever a new model has to be designed from scratch, we recommend to first use *CellDesigner* to define the model entities. Afterwards, the model can be loaded into BioPARKIN for “fine-tuning”, i.e. to identify parameters and modify mechanisms.

2.2 Input and Output of Experimental or Simulated Data

A tab-separated value pseudo-file format (with ending “txt” or “csv”) was chosen to save experimental data or simulation results. It uses tabulators to separate fields and needs to have the structure given in Table 1. The first column has to be named “Timepoint [unit]” with “unit” defining the actual time unit of the measured (or simulated) data. The timepoint need not to be ordered or to be unique. Of the following columns, every column ending with “[unit]” or “[]” is treated as data of a specified substance (species). The headers of these data columns must correspond to species IDs within the model so that BioPARKIN can match model entities and data.

If a column called “SD” is preceded by a data column it holds the standard deviations of the measurements in the preceding column.

Finally, columns with arbitrary names can be appended at the end. As long as they do not contain square brackets, they are effectively ignored by BioPARKIN. This can be helpful e.g. to keep track of patient IDs within files, or to add any comments.

BioPARKIN not only reads files formatted this way but uses the same format to save simulation results to disk.

3 Graphical User Interface - Overview

This section gives an overview of BioPARKIN’s graphical user interface (GUI). When starting BioPARKIN, the user is given a window with a menu bar and icons at the top. Within the menu bar one can amongst others perform calculations and open the *ODE Viewer*. The *ODE Viewer* shows the generated differential equations of the model in different forms, i.e., with component IDs or names, with reaction abbreviations, or in full notation. Within the icons one can amongst others switch to the *Results Window*. If an error or a warning occurs, a grey cloude appears on the right hand side of the icon list. Once you click on this cloud, a window with more specific information pops up, and the cloud turns into a sun.

The main part of the window is taken up by two tabs that show the constituent parts of the current model (*Model Overview*) and the simulation control (*Workbench*). After computations, a second window, the *Results Window*, shows up.

3.1 Model Overview

This tab is devided into three parts. Under *Model List* on the left, the currently opened model is shown. In future it will be possible to open several models at the same time and quickly switch between them, perform simulations, and compare results.

In the center, under *Entity Tree*, there is a tree-like view of all the model’s constituents, i.e. compartments, species, reactions, etc. Together they define the entire model. When clicking on one entity from the list, information about its properties are shown on the right.

Under *Entity Details* the attributes of the currently selected entity are shown, together with their current values. Details that are not modifiable by the user are greyed out.

Currently, the *Model Overview* tab is meant to give an overview of the constituents of the opened model. A future update of BioPARKIN will allow to build and edit a model—e.g. add and remove species and reactions—from within the Model Overview tab. Due to the *SBML* principle of interoperability between software tools, it is easily possible to use other programs to build and change a model. At present, whenever the user wants to modify species, reactions, rules, or events, we recommend to edit the model in *CellDesigner* and to reload it into BioPARKIN. However, changes in names (of species, parameters, rules, or events), in formulas (assignment rules, reaction rates), and event specifications can already be done here and will be handeled and displayed consistently throughout the rest of the software.

3.2 Workbench

This tab hosts the settings and controls to start simulations (including the generation of modifiable and saveable figures and tables), to perform detailed sensitivity analysis or to identify parameters :

- Adjust global settings for the integrator (these changes will impact all computations)

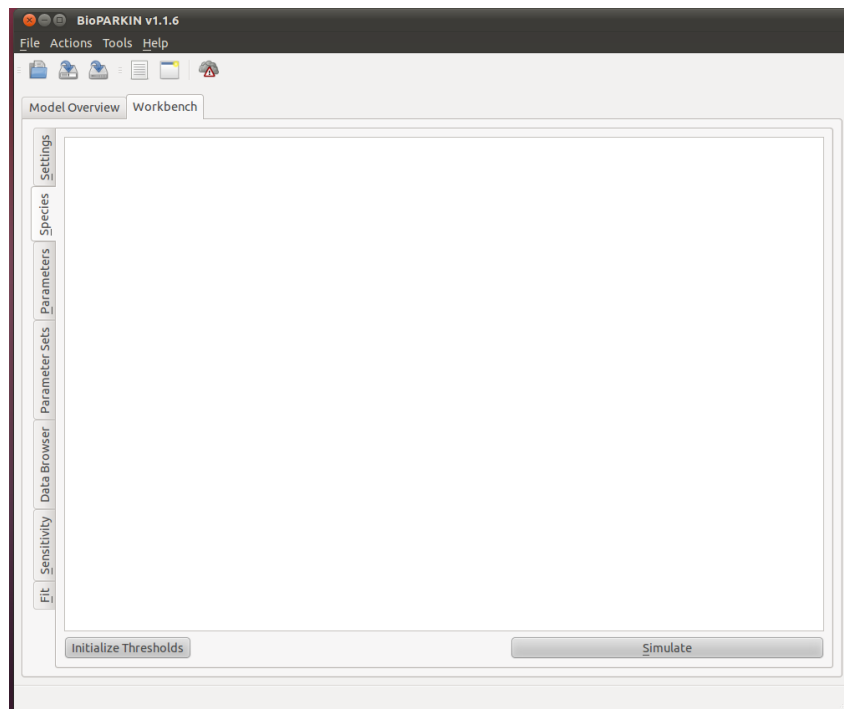
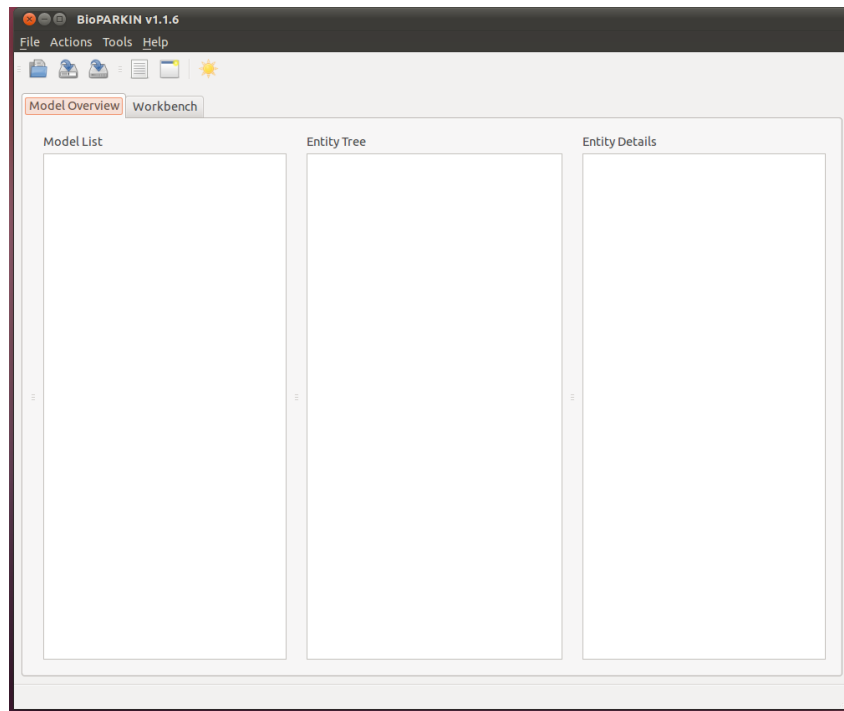


Figure 1: Starting window of BioPARKIN. Top: *Model Overview* tab. Bottom: *Workbench* tab.

- Modify names, value or thresholds of species, and select species that are treated as boundary condition
- Modify names, values or thresholds of parameters
- Manage parameter sets
- Manage experimental data (Load, select, shift and save multiple data sets)
- Select parameters and species taken into account in the sensitivity analysis
- Select parameters that are identified based on experimental data

These tabs will be explained in more detail in Section 4 of the manual.

3.3 Results Window

In this window, which opens automatically after performing a simulation, a sensitivity analysis or parameter estimation, different tabs show the results of the numerical computations. These tabs are explained in detail in Section 4. On the bottom of this window, the user can change the view between tab mode and window mode, close the current tab or close all tabs. Note that the results are lost if a tab is closed without saving the data before.

4 Graphical User Interface - Usage

This section illustrates the use of BioPARKIN in step-by-step descriptions from the perspective of a user, aiming to guide through the features of the software.

4.1 Model Overview: Working with a Model

When BioPARKIN is started, in the upper left corner, click on *File/Open* to get access to your home directory. Choose the desired *SBML*-file and click on *Open*. The file name now appears in the model list. A short-cut icon for this process is available right below *File*. If you worked previously with that model, it also appears in a list of at most 6 model files which opens when you click on *File*. In this case, you can directly click on the desired *SBML*-file without opening the file browser.

The column *Entity Tree* in the middle now shows all model components: Compartments, species, reactions, parameters, rate rules, assignment rules, and events. By clicking on the plus sign or arrow in front of one of these components, the tree opens up and shows the list of all objects belonging to a specific component, e.g. the list of all species. A click on one of these objects reveals its properties and property values in the right column *Entity Details*. Note that this list is not the complete *SBML* entity catalogue. BioPARKIN supports only a selection of entities essential for the numerical computation tools. Modifiable property values are displayed in black, fixed ones are grey.

- Compartments are characterized by their volume (size). If the compartment ID occurs in the equations, it is replaced by the volume during the computations. The compartment name is optional.
- The details of the species appear as they have been specified in the software that was used to build the model. The initial quantity can be changed in *Workbench/Species*. To edit other properties, a different software, e.g. *CellDesigner*, has to be used.

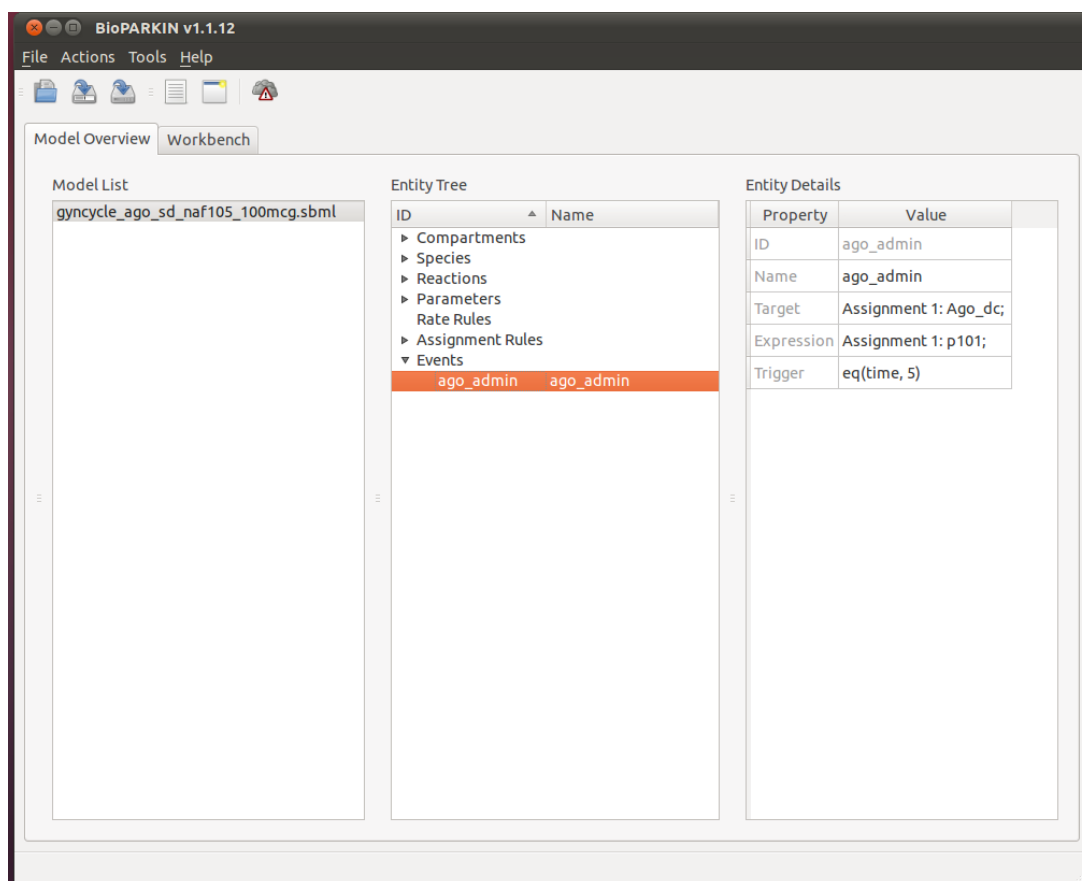


Figure 2: *Model Overview* tab with a loaded model.

- The reaction list contains the mathematical descriptions of all interactions between species, which build up the right hand side of the ODE system. In particular, under *Entity Details*, the *Math* property describes the corresponding reaction rate, i.e. the rate of change in amount or concentration of a species from *Reactants* to *Products*. This expression has to be formulated in *MathML*², and uses the IDs of the involved parameters and species.
- As for the species, the details of the parameters appear as constructed in other software. Parameter values can be changed in *Workbench/Parameters*.
- Rate rules are used to modify species without introducing new reactions. A rate rule sets the rate of change of a species to the value determined by the formula in *Math*. Most models, however, do not use this possibility such that the rate rule list is often empty.
- Assignment rules are used as placeholder for certain algebraic expressions. (In contrast to *Algebraic Rules*, which are not supported by the current BioPARKIN release, *Assignment Rules* do not represent constraints on the systems.) The object defined by an assignment rule can either be constant (when it depends only on constant entities, e.g. parameters) or dynamic, i.e. time-dependent (when it depends on other dynamic entities, e.g. species).
- Events take place at breakpoints, $t_i > t_0$, defined by the *Trigger* eq(time, t_i). Here “time” is the time variable, i.e. the variable of integration. At the time of an event, new values can be assigned to species and parameters. These assignments are defined by a *Target* specifying the species or parameter ID, and an *Expression* that specifies the value to be assigned. BioPARKIN does not yet support state-dependent events.

If changes are made, the modified model can be saved by clicking on *File/Save* or *File/Save as*, respectively. For these commands, short-cut icons are available in the icon list as well.

In addition to the standard SBML view, BioPARKIN offers the possibility to view the generated differential equations by clicking on the 4th icon in the upper left corner. The ODEs are displayed both in terms on IDs as well as names, with reaction abbreviations or in full notation.

You can close a model by clicking on *File/Close*. In contrast to *Close*, *Quit* will end the program!

4.2 Workbench: Settings

The tab *Workbench/Settings* contains a number of specifications that are required to simulate the model.

4.2.1 General Settings

For parameter identification, two different Gauß-Newton codes can be used, either *Gauß-Newton (NLSCON)* or *Gauß-Newton (PARKIN)*. PARKIN is a variant of the standard NLSCON with a different damping strategy. The user might wish to test and compare both variants.

If experimental data are loaded, which shall be included in the simulation plots, click the corresponding check box.

²SBML uses a subset of MathML 2.0 standard. www.w3.org/TR/MathML.

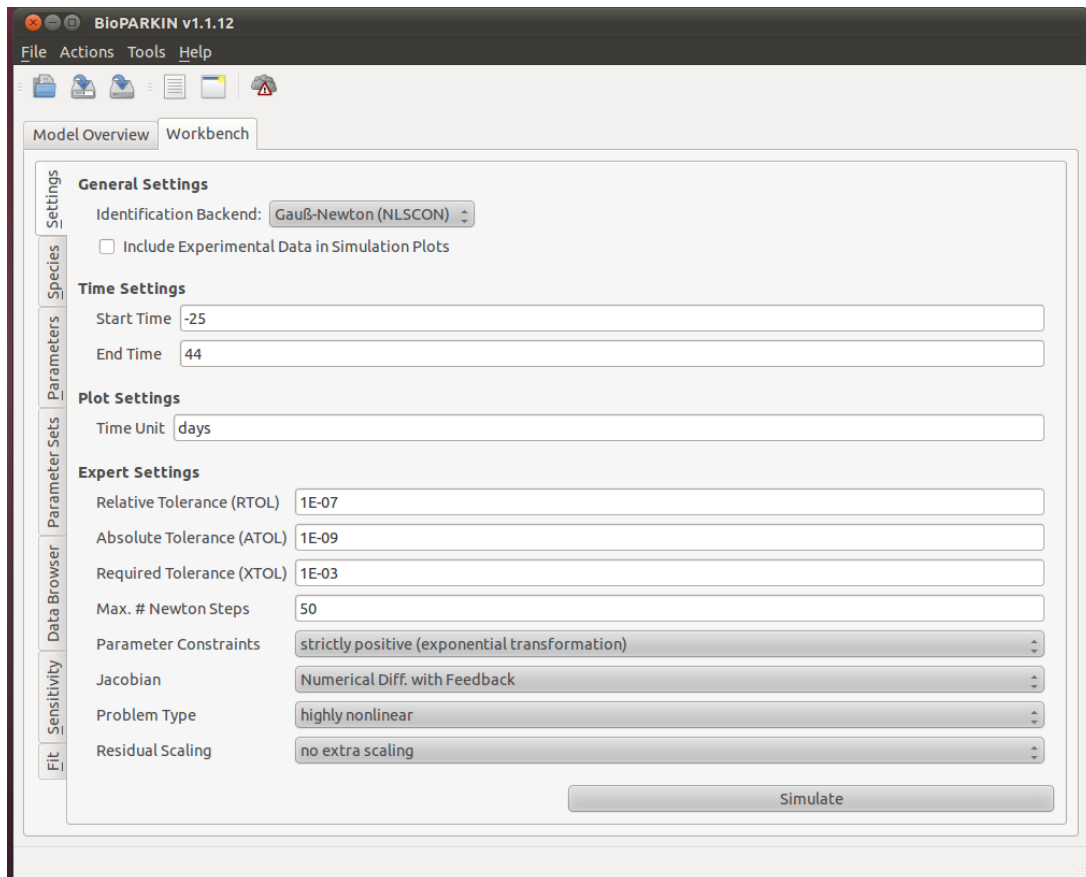


Figure 3: *Workbench/Settings* tab.

4.2.2 Time Settings

Start and end time for the simulation have to be defined. Note that time is allowed to take negative values. In particular for autonomous systems, where the right hand side of the ODE system does not explicitly depend on time, this does not change the solution but only the labeling of the time axis.

Note that time shifts can also be included when loading experimental data in the *Data Browser*.

4.2.3 Plot Settings

Here the user can directly edit the string of the time unit which is displayed on the x-axis of any simulation plot.

4.2.4 Expert Settings

Numerical integration of the ODE system is performed with LIMEX, a Linearly IMplicit Euler method with EXtrapolation [5]. LIMEX is suitable for stiff differential equations as they often occur in chemical reaction kinetics. The values *RTOL* (relative tolerance) and *ATOL* (absolute tolerance) are controlling the adaptive time step selection of the integrator. For numerical details, we refer to [2]. There are model systems that are only solvable with very small absolute tolerance, e.g the Zhabotinski-Belousov reaction described in the aforementioned book. Negative solutions for species which are expected to stay positive can be an indicator for small absolute tolerance requirements. Note, however, that a too small tolerance could cause the integrator to interrupt before integration is completed because the desired tolerance might not be achievable.

Further settings refer to the damping strategy of the Gauß-Newton method during parameter identification. Some of these settings only apply to *Gauß-Newton (NLSCON)*. If *Gauß-Newton (PARKIN)* is selected, which works much more in an automated way, the respective settings are greyed out.

- The tolerance *XTOL* refers to the the desired relative accuracy of the estimated parameters with respect to affine covariance [1].
- *Max # Newton Steps* is the maximum allowed number of iterations in the Newton method. When this number is reached, the iterations are stopped even if convergence has not yet been achieved.
- Sometimes parameters might be required to be positive. In this case, chose *strictly positive* as *Parameter Constraints*. At the moment, this option can only be applied to all parameters or none, but this will be improved in a future release.
- The *Jacobian* can be computed in three different ways, either by computing difference quotients (*Numerical Diff. with Feedback* or without feedback) or by solving the variational equation (*Variational Equations*) as described in A.4. It is well known that numerical differentiation is numerically unstable, but in this version of BioPARKIN it is more likely to be faster, especially for large systems. Numerical differentiation with feedback, however, tries to maximize the achievable precision by using numerical safeguards. Note that the method *Gauß-Newton (PARKIN)* only uses variational equations.
- Problems involving chemical reaction kinetics are usually nonlinear. Different degrees of nonlinearity (mildly/highly/extremely) require different damping strategies in the Gauß-Newton iteration. Unfortunately, the equations themselves do not reveal their degree of nonlinearity. We recommend to change the *Problem Type* whenever NLSCON does not converge, e.g. in case of too small damping factors. PARKIN uses a fixed damping strategy independent of the degree of nonlinearity.

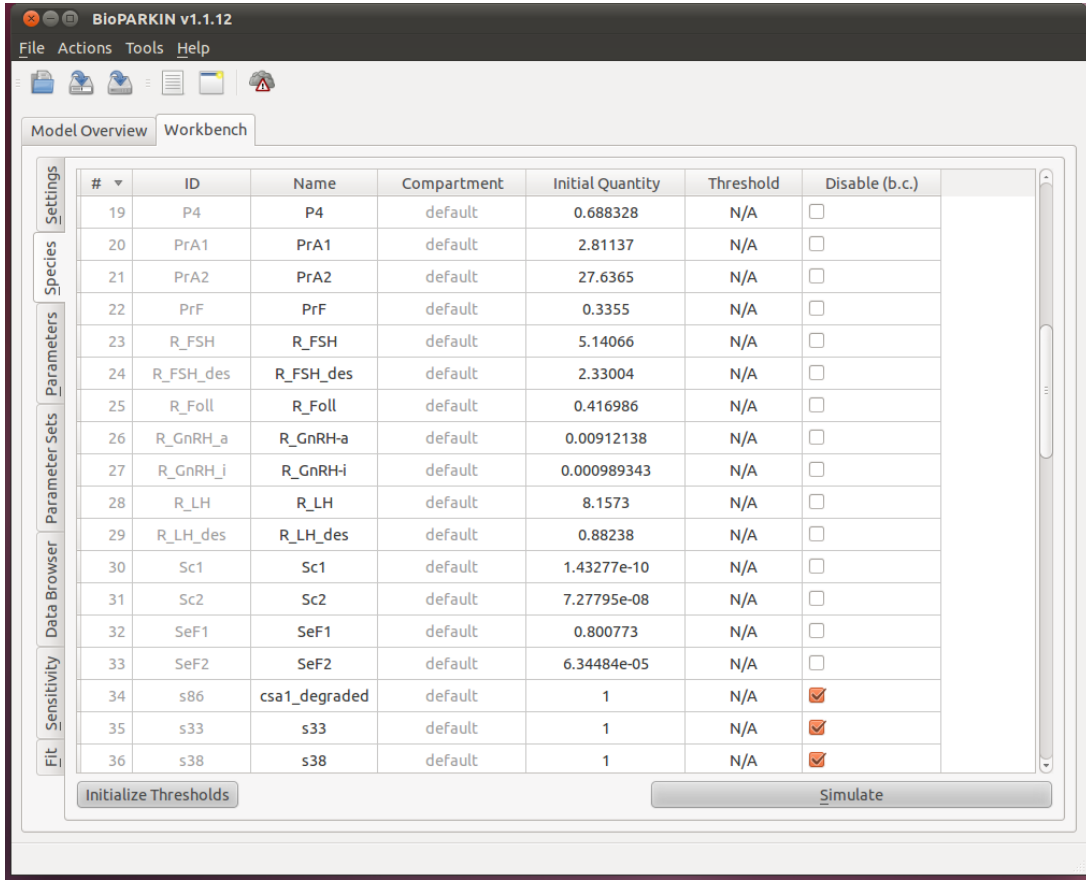


Figure 4: *Workbench/Species* tab.

- Numerical experiments in [13] have shown that it might sometimes be useful to scale the residuals during the Newton-iteration. *Residual Scaling* should by default be set to *no extra scaling*. If the algorithm cannot be applied successfully, it could be an option to perform relative or absolute scaling. This only applies to NLSCON. Note, however, that residual scaling is not covered by theory.

4.2.5 Simulation

A click on the button *Simulate* in the lower right corner starts the numerical integration of the ODE system with the current settings. Results are displayed in the *Results Window*, which is explained in detail in Section 4.9.

4.3 Workbench: Species

On the tab *Workbench/Species* you find the list of all species with properties and property values. The rows can be sorted alphabetically or numerically by clicking on the header of a specific column.

Compared to the species list in the tab *Model Overview*, an additional column *Threshold* appears. These entries are important for sensitivity analysis and parameter identification since they account for different scales, caused by e.g. different physical units of the species, compare Section A. By clicking on *Initialize Thresholds*, the values are set to the machine accuracy ε . However, a warning pops up which informs the user that this might not be the right choice.

In the *Species* tab, species' names and initial quantities can be edited. The column *Disable (b.c.)* has a check mark whenever the property *Boundary Condition* (=b.c., predefined SBML property name) for the species in the *Model Overview* tab is *true*. In this case, a species is not integrated and handled as zero in the remaining differential equations, which is usually desirable for, e.g., degradation products. If *Boundary Condition* is set *true* by mistake, this results in a different model and thus a different simulation result.

A click on the button *Simulate* in the lower right corner starts the numerical integration of the ODE system with the current settings. Results are displayed in the *Results Window*.

4.4 Workbench: Parameters

The tab *Workbench/Parameters* contains the list of all parameters with properties and property values. The rows can be sorted alphabetically or numerically by clicking on the header of a specific column.

The *Threshold* value gives a lower bound for the absolute value of each parameter. Again, these entries are important for sensitivity analysis and parameter identification since they account for different scales caused by, e.g., different physical units of the parameters. By clicking on *Initialize Thresholds*, the values can be set to machine accuracy ε . However, a warning pops up which informs the user that this might not be the right choice.

In addition, the list contains variables defined by assignment rules. Instead of a numeric value, the column *Value* contains the word *Assignment Rule* for these variables.

A click on the button *Simulate* in the lower right corner starts the numerical integration of the ODE system with the current settings. Results are displayed in the *Results Window*.

4.5 Workbench: Parameter Sets

When the user is working on a model or exploring its parameter space, parameter values are usually changed quite often. The parameter sets feature of BioPARKIN tries to accommodate this behaviour.

In the tab *Parameter Sets* BioPARKIN offers the possibility of bookkeeping several parameter sets for a single set of equations. Each parameter set includes all parameters and their values. Parameter sets are displayed side-by-side in the tab and can easily be duplicated or removed.

Upon opening a SBML model, an "Original" parameter set is created that allows the user to revert back to the initial parameter values at any time. Another set is created automatically, called "Initial Guess". This is meant to be used in the context of parameter identification. The results of an identification run are also put into a parameter set and, thus, can quickly be plotted, compared with measurements, etc.

All simulations are performed with the parameter values in the set labeled as *Active*. Whenever parameter values are changed in the *Model Overview* or in *Workbench/Parameters*, these changes are displayed in the active set. Vice versa, changes made in the active parameter set are displayed in the other tabs as well. If a set is labeled as *Selected*, the buttons in the lower right corner can be applied. A set can

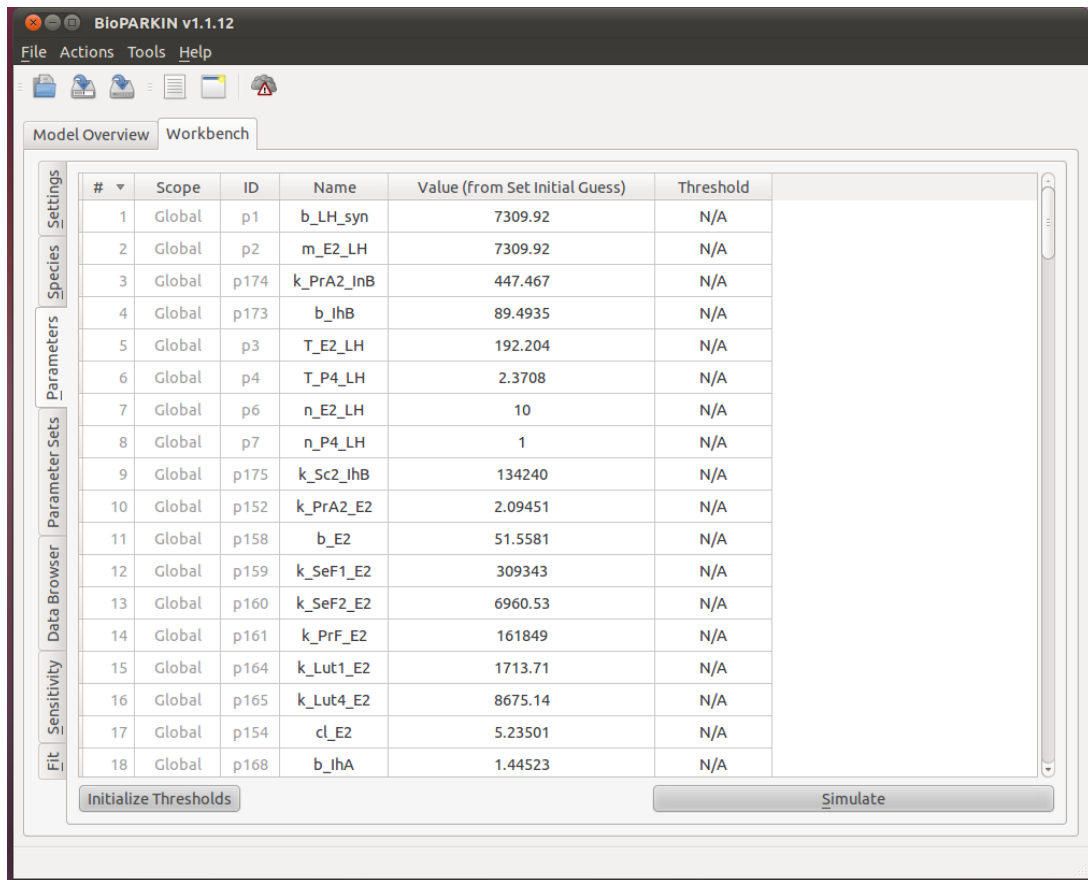


Figure 5: *Workbench/Parameters* tab.

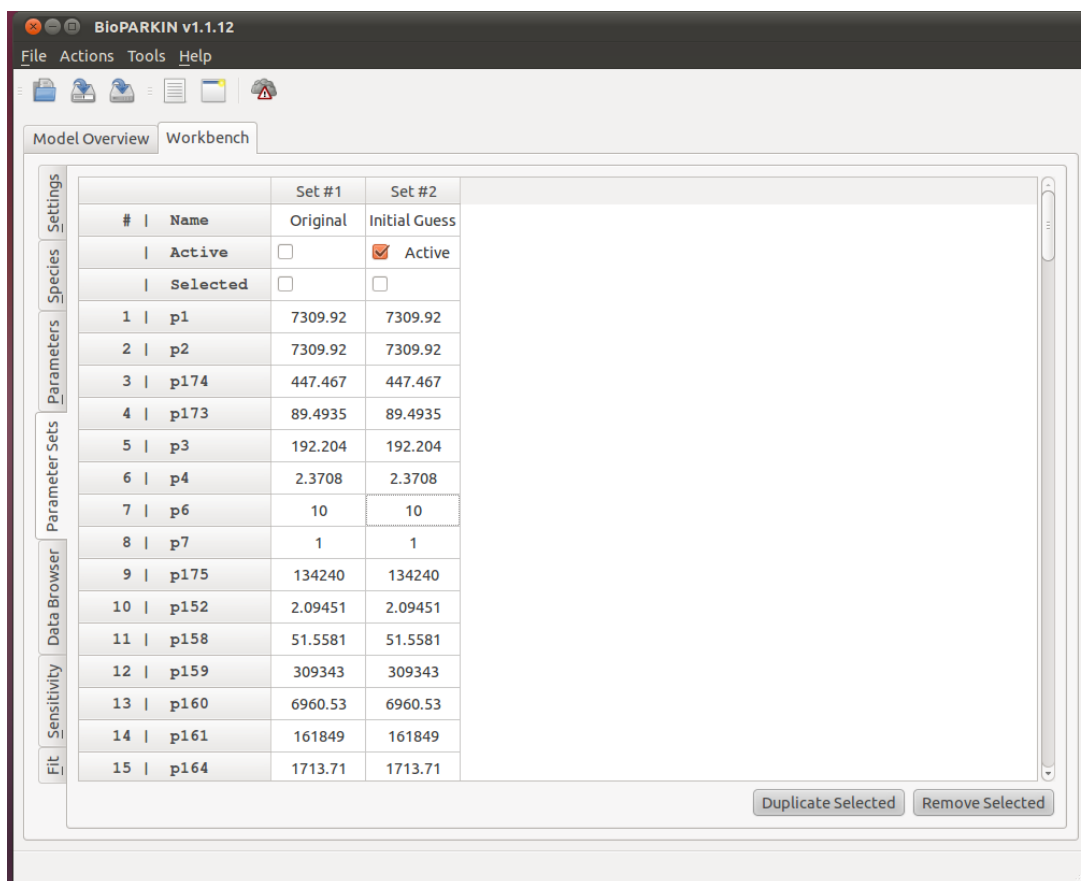


Figure 6: *Workbench/Parameter Sets* tab.

be removed by clicking on the corresponding button *Remove Selected*. By a click on the button *Duplicate Selected* in the lower right corner, a selected set can be duplicated to save intermediate parameter sets before further modification. If no set is marked as selected, then the active set is duplicated.

When a model is saved, all parameter sets are stored. They are reloaded when the model file is opened again. If the SBML-file is opened with a different software, e.g. Copasi, only that parameter set is loaded which was labeled as active at the timepoint of saving.

4.6 Workbench: Data Browser

In the tab *Workbench/Data Browser* the user has the possibility to browse through the file system (click on *Browse*) and import data files (click on *Import*), provided they are in the format described in 2.2. The loaded data are presented column-wise in a table. They can be plotted in a separate *Results Window*, which comes up if one clicks on *Plot*. Check boxes on top of each column allow for a selection of species for plotting or parameter identification. Selected columns can be saved as new table (button *Save As...*). A click on the button *Plot* in the lower right corner opens the *Results Window* with a plot of the data in the table currently shown on the left.

On the right hand side of the table, there are two tabs, named *Information* and *Actions*. The tab *Information* displays information on the file and the data. In the tab *Actions*, the user can shift data points in time by entering the desired timeshift and clicking on *Timeshift Data*. This tool is especially useful for periodic systems as, e.g., the human menstrual cycle, where changes in parameter values or initial values can shift the periodic solution along the time axis.

BioPARKIN offers the possibility to include data from several files. On the right hand side of the header at the bottom of the last data table, there is a plus sign. A click on this sign opens the initial data browser view with the two buttons *Browse* and *Import* to include further data files.

Data sets are selected if they are checkmarked in the lower left corner of the corresponding table header. Several data sets can be plotted in one figure by clicking on *Plot All Data Sets* or *Plot Selected Data Sets*. Data sets are closed by clicking on the cross in the header below the table.

4.7 Workbench: Sensitivity

The tab *Workbench/Sensitivity* contains a list with all parameters on the left, and a list with all species on the right. In the parameter list, the user can click check boxes to select specific parameters for which sensitivities shall be computed. A similar selection can be made in the species list. Note, however, that this selection is mainly for plotting purpose, not for saving computing time. If the Jacobian is computed by solving variational equations, (see *Workbench/Settings*) the computing time does not depend on the number of selected species or parameters since the full ODE system has to be solved in any case. If numerical differentiation is chosen, the computing time depends linearly on the number of selected parameters. If *Gauß-Newton (PARKIN)* is selected as identification backend in *Workbench/Settings*, sensitivities can only be computed if threshold for parameter values have been specified in *Workbench/Parameters*. For *Gauß-Newton (PARKIN)*, which is the default setting, the following rule applies:

Sensitivities cannot be computed until thresholds for species (*Workbench/Species*) and parameters (*Workbench/Parameters*) have been defined!

The sensitivity of a species with respect to a parameter depends on time. Thus, to start the computation the user has to specify the time. There are three possibilities:

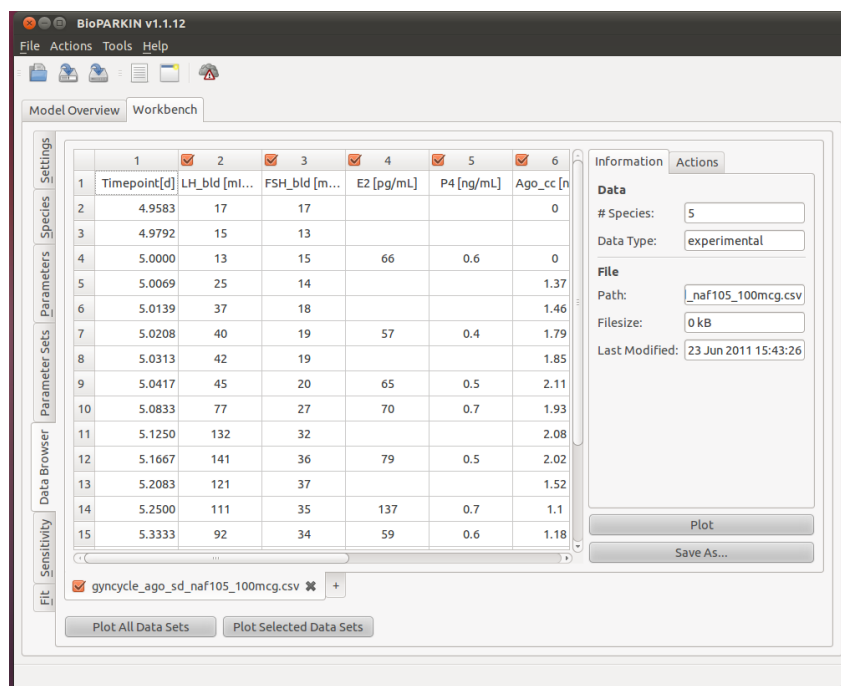
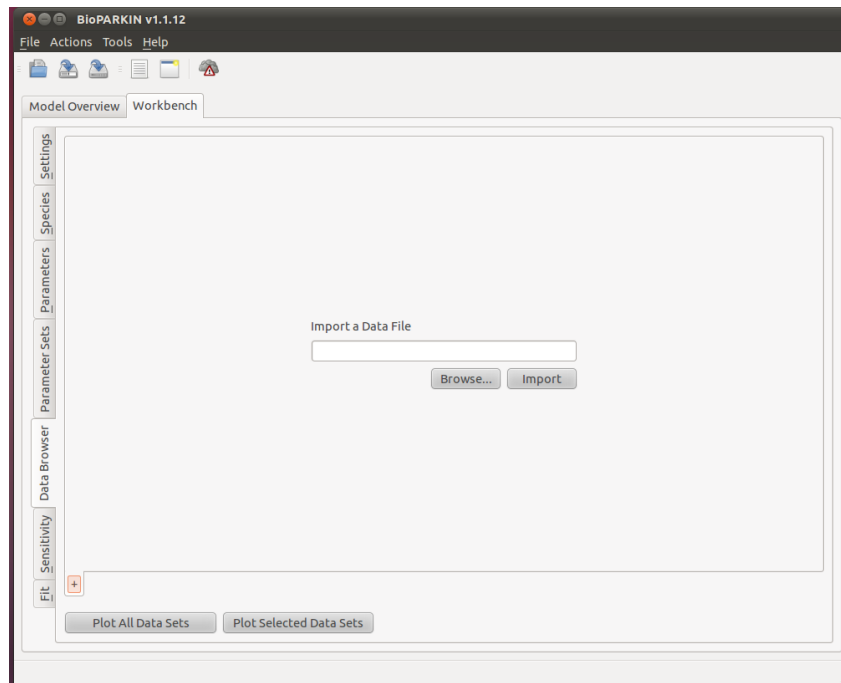


Figure 7: *Workbench/Data Browser* tab. Top: Before loading data. Bottom: With loaded data.

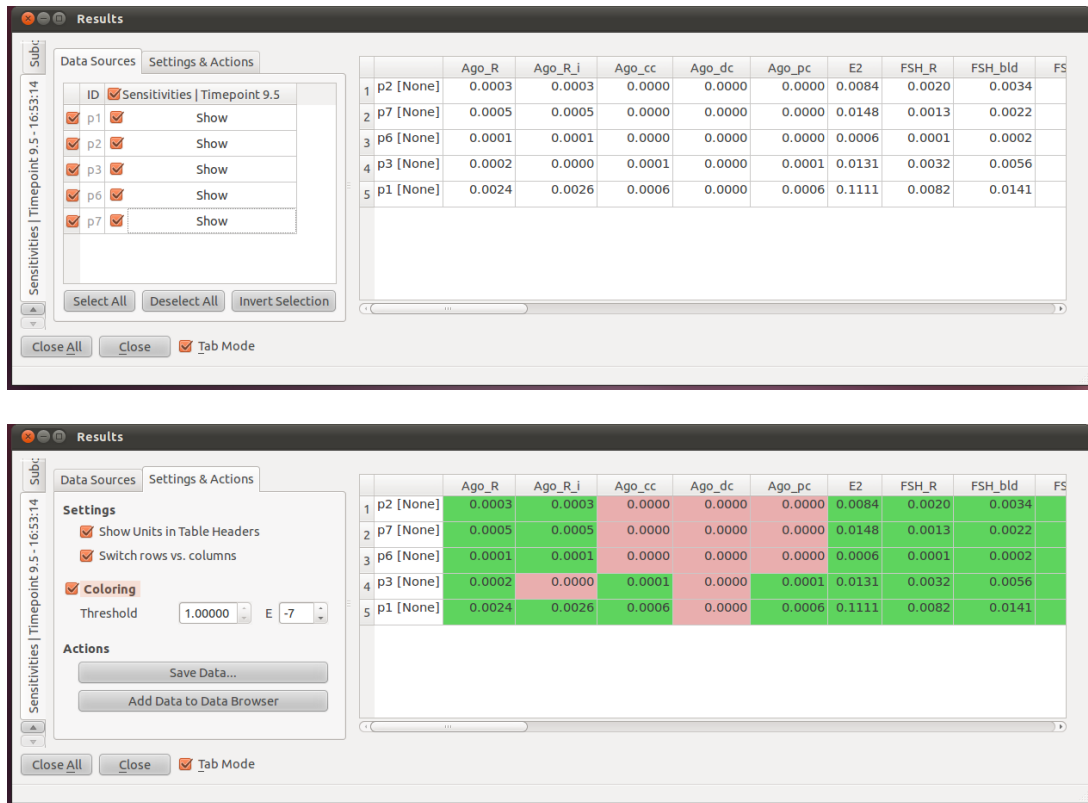


Figure 8: *Results Window*: detailed sensitivity analysis (Matrices).

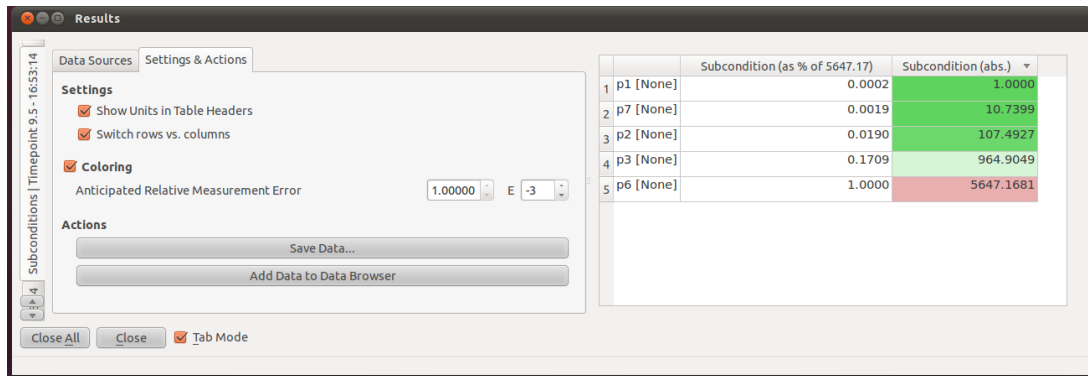
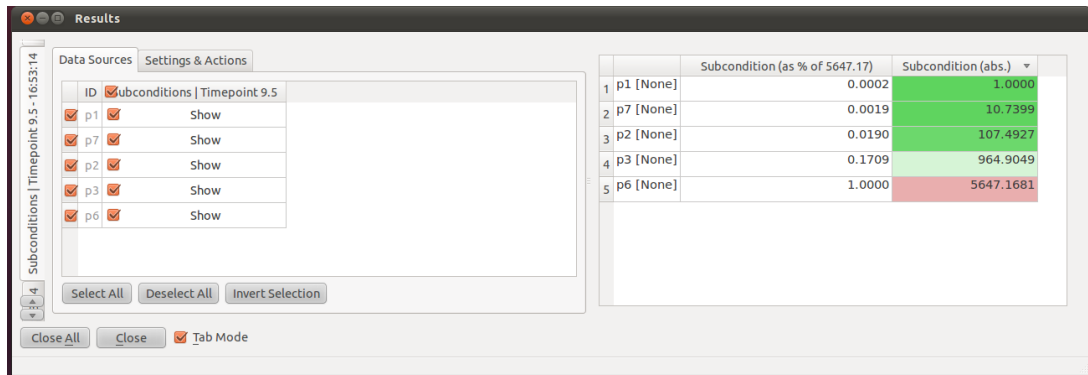


Figure 9: *Results Window*: Detailed sensitivity analysis (Subconditions).

1. The sensitivities are computed for the complete time interval I specified in the tab *Workbench/Settings*. The sensitivities over time are displayed in the *Results Window*.
2. The sensitivities are computed at user-specified time points.
3. The sensitivities are computed at all time points where measurement values of certain species are available.

Case 1 is handled via the Button *Sensitivity Overview* below the two tables. After a click on *Compute Sensitivities*, the sensitivities are calculated over the whole time interval specified in *Workbench/Settings* and the results are shown in the *Results Window*, both as table as well as time plot.

Case 2 and 3 are handled under *Detailed Sensitivities*. One can choose between *Get Timepoints from Data* or specifying time points by hand. Time points can directly be edited in the window below *Timepoints (space-separated)* or calculated from *Start Time*, *End Time*, and *#Intervals* or *Interval Size*. For every timepoint, two separate tabs open in the *Results Window*. One tab shows the sensitivities of the selected species with respect to the selected parameters at that specific time point. The other tab contains the subcondition numbers. For more details we refer to 4.9.2. To keep an overview over all timepoints, it is recommended not to use too many of them at the same time.

4.8 Workbench: Fit

If experimental data are loaded, BioPARKIN can be used for parameter identification. That means, the underlying algorithm NLSCON determines the subset of parameters that can actually be identified from the given data and changes these parameter values to obtain the best fit to the data in a least squares sense, compare A.5.

In the tab *Workbench/Fit* the user can select parameters for optimization by clicking on the checkbox in the column *Estimate Value*. It is recommended to first run a sensitivity analysis with the desired parameters and to select only these parameters who are indeed identifiable. A click on *Identify Parameters* in the lower right corner starts the Gauß-Newton method. The final values, which are displayed in the tab *Par.Identification* in the *Results Window*, are stored as new set in *Workbench/Parameter Sets* and marked as active set such that the next simulation would use these new values.

4.8.1 NLSCON

The output of this subroutine is displayed in the BioPARKIN command shell. The settings for NLSCON are followed by a table of iteration steps containing

1. The iteration step number
2. The value of $\|F(p)\|_2/M$ (Normf)
3. Weighted update norm $\|D^{-1}\Delta p^k\|_2$ (Normx)
4. The damping factor λ_k (Damp.Fct)
5. A monitor for rank one Newton update steps (not used at the moment)
6. The rank of the Jacobian in the current iteration step

The NLSCON options FCSTART and FCMIN are determined by the user-defined non-linearity of the problem (Workbench/Settings), whereas COND is determined as $1/\text{RTOL}$. Besides other criteria, the iteration terminates as soon as $\text{Norm}x < \text{XTOL}$.

After NLSCON has come to an end, the iterates p^k are displayed.

4.8.2 PARKIN

Amongst other things, the command shell displays internal scaling values for measurements, which are chosen automatically by PARKIN. These values correspond to the measurement tolerances δy_i (compare Section A.5). Moreover in each iteration, the following output is produced:

1. The iteration step number (IT)
2. The norm of the update step $\|\Delta p^k\|_2$ (LEVELX)
3. The damping factor λ_k (RELFC)
4. The rank of the Jacobian in the current iteration step (RANK)
5. The weighted least squares norm $\|F\|_2$ (LEVELF)
6. The largest subcondition number of the parameters to be estimated (SUBCOND(J))
7. The number $|r_{11}|$ from the QR decomposition (SENS). A large value indicates large sensitivity.
8. The inkompatibility factor κ (KAPPA). The identification result only makes sense if $\kappa < \frac{1}{2}$. If κ is larger, especially $\kappa > 1$, model and data do not fit together, and the result should be discarded.

There two output lines. The first line contains the values of some items at the beginning of a Newton iteration, the second line contains the values of some items at the end, depending on when they are computed. In addition, the results of a rank optimization step are displayed. For details we refer to [1].

4.9 Results Window

Every performed calculation leads to a new tab in the *Results Window*. There are two different types of tabs, corresponding to two presentation types for the results: figures (data plot) and tables (data table).

In both tab types, the number of displayed species can be controled by clicking on *Show* or *Hide* in the subordinated tab *Data Sources*. There are shortcuts for this selection process via the buttons *Select All*, *Deselect All*, and *Invert Selection*.

4.9.1 Data Plots

There are three types of data plots, that are created as new tab in the *Results Window* after the corresponding calculation has been performed:

1. Simulation (Plot)
2. Experimental (Plot)
3. Sensitivities (Plot)

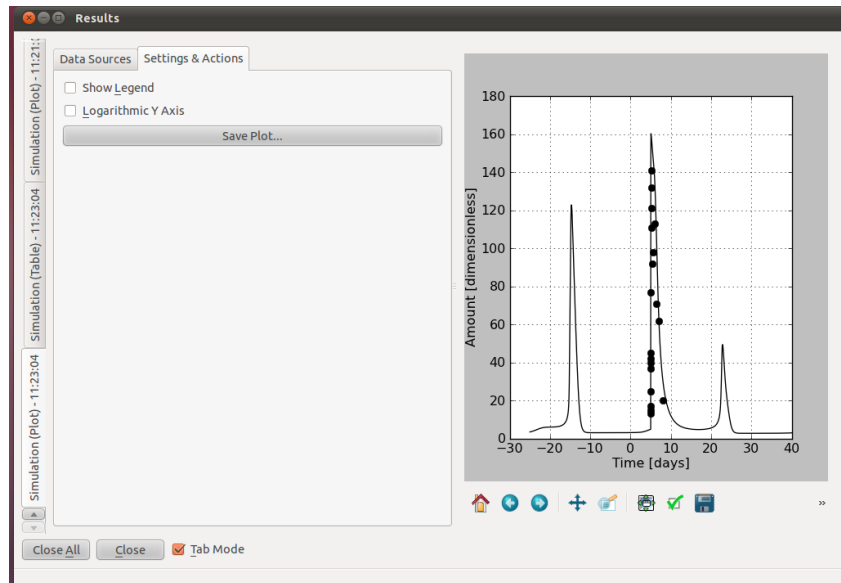
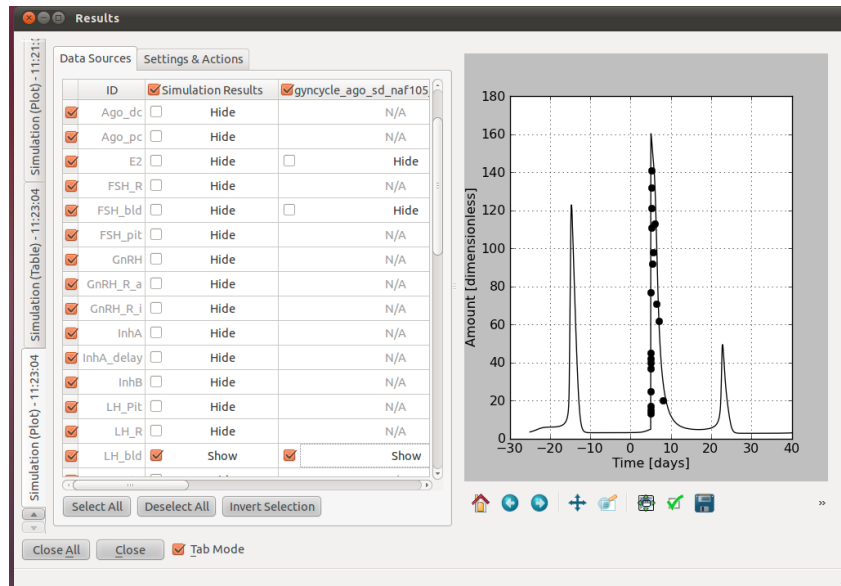


Figure 10: *Results Window: Simulation (Plot).*

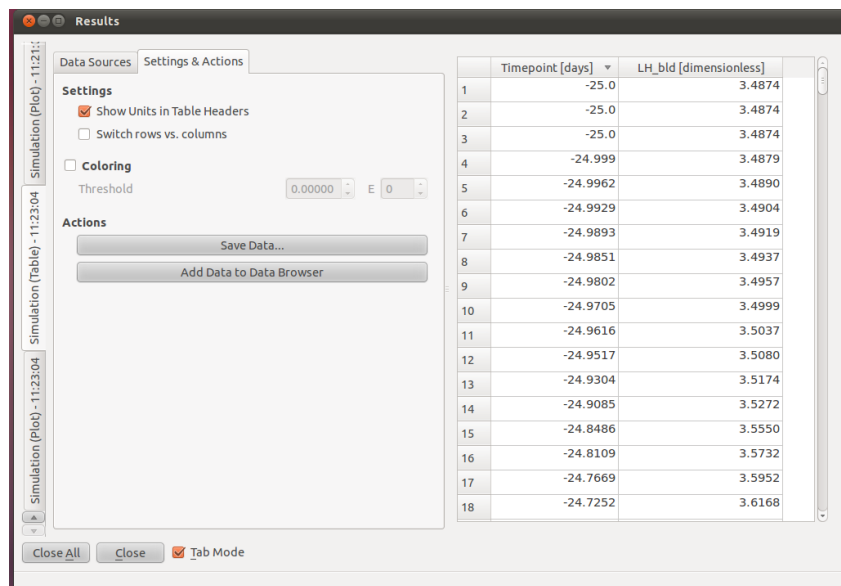
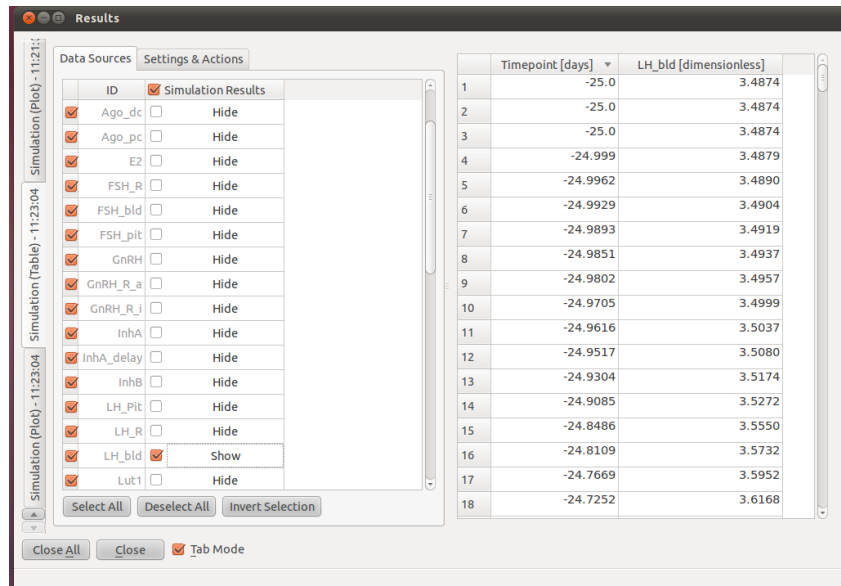


Figure 11: *Results Window: Simulation (Table).*

Within these tabs, the figure is on the right, and two subordinated tabs named *Data Sources* and *Setting & Actions* are on the left. In the tab *Settings & Actions*, the user can switch on a legend, change the scaling of the y-axis between linear and logarithmic, and save the plot in a number of predefined formats (e.g. png, jpg, eps, or pdf). In addition, below the figure on the right there are icons for panning and zooming in and out of the graphic.

Note that typical machine precisions are about $1e - 16$. If *Simulation(Plot)* in the *Results Window* shows solution curves below machine precision, there are two possible interpretations:

1. The species is in fact zero, so this has no effect on the correctness of the rest of solution. Do not panic:)
2. It could be that the current model indeed induces such small values. In this case the user should think about introducing different physical units to avoid this effect. The solution, however, is still within the chosen tolerance (*ATOL*, *RTOL*).

Explicitly forcing species to zero is not advisable, since these two cases could not be distinguished by the integrator. If a species is zero or very small, it may happen that the sensitivity computed with numerical differentiations gets large due to numerical cancellation, whereas the sensitivity computed by solving the variational equation is within the desired tolerance, compare Fig.12.

4.9.2 Data Tables

There are three types of data tables, that arise as new tab in the *Results Window* after the corresponding calculation has been performed:

1. Simulation (Table)
2. Sensitivities
3. Subconditions
4. Par.Identification

Within these tabs the table is on the right, and two tabs named *Data Sources* and *Setting & Actions* are on the left. The tab *Settings & Actions* offers the possibility to change the layout of the table and to save the data. In addition, it is possible to add computed data from the table to experimental data (*Add Data to Data Browser*). In this case, a new tab named *Pseudo-Experimental Data* is added in *Workbench/Data Browser*. Moreover, values in the table can be colored by activating the check box *Coloring*. The user can specify a threshold below which the table entries are colored red, whereas larger values are colored green.

For single time points, parameters are ordered according to their subcondition number. By default, subconditions below $1/RTOL$ are colored green, which means that the corresponding parameters are identifiable, whereas larger values are colored in red. It might happen, however, that the relative measurement error of experimental data is larger than *RTOL*. In this case, fewer parameters are identifiable, which become visible by the changing coloring if the threshold *Anticipated Relative Measurement Error* is increased.

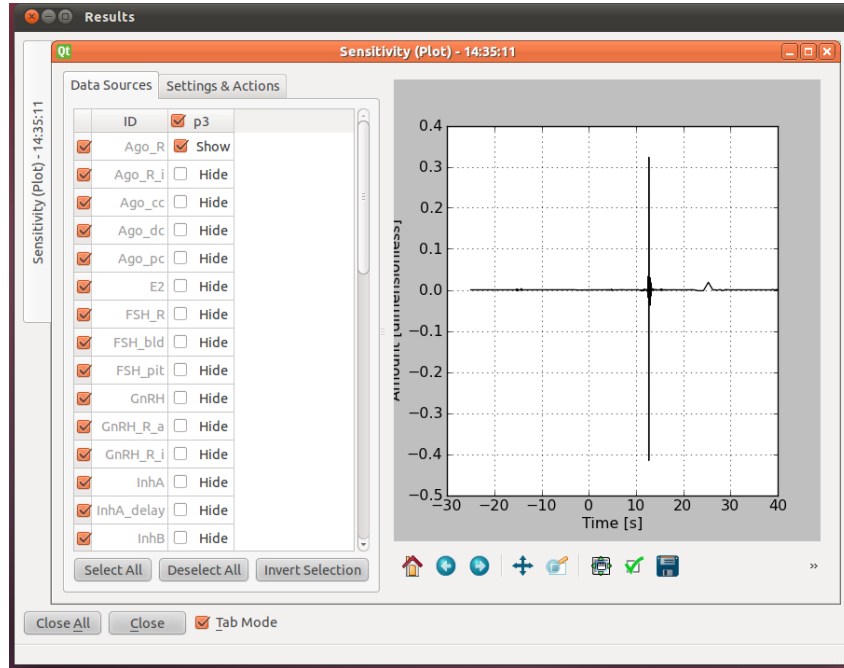
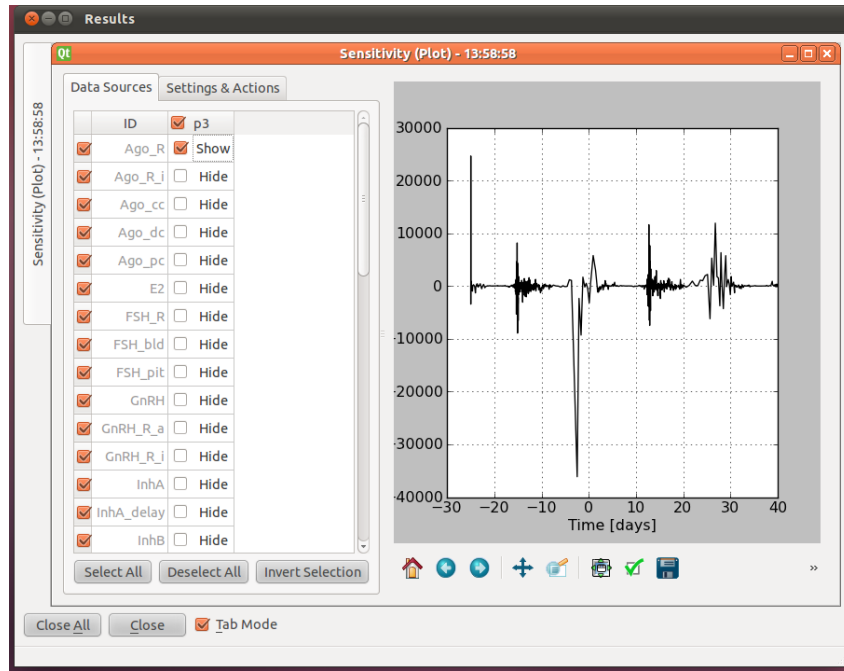


Figure 12: *Results Window* with sensitivity plots for the species `Ago_R` with small concentration. Top: Sensitivities computed by numerical differentiation. Bottom: Sensitivities computed by variational equations.

References

- [1] P. Deuffhard. *Newton Methods for Nonlinear Problems – Affine Invariance and Adaptive Algorithms*. Number 35 in Springer Series in Computational Mathematics. Springer, 2004.
- [2] P. Deuffhard and F. Bornemann. *Scientific Computing with Ordinary Differential Equations*. Number 42 in Texts in Applied Mathematics. Springer-Verlag, 1 edition, 2002.
- [3] P. Deuffhard and A. Hohmann. *Numerical analysis in Modern Scientific Computing – An Introduction*. Number 43 in Texts in Applied Mathematics. Springer-Verlag, 2 edition, 2003.
- [4] P. Deuffhard and U. Nowak. Efficient numerical simulation and identification of large chemical reaction systems. *Berichte der Bunsengesellschaft für physikalische Chemie*, 90(11):940–946, 1986.
- [5] P. Deuffhard and U. Nowak. Extrapolation integrators for quasilinear implicit ODEs. In P. Deuffhard and B. Engquist, editors, *Large Scale Scientific Computing*, pages 37–50. Birkhäuser, 1987.
- [6] A. Funahashi, N. Tanimura, M. Morohashi, and H. Kitano. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks, 2003.
- [7] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI - a complex pathway simulator. *Bioinformatics*, 22(24):3067, 2006.
- [8] M. Hucka, S. Hoops, S.M. Keating, N. Le Novère, S. Sahle, and D. Wilkinson. Systems biology markup language (sbml) level 2: structures and facilities for model definitions. 2008.
- [9] N. Le Novère, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M.I. Aladjem, S.M. Wimalaratne, et al. The systems biology graphical notation. *Nature biotechnology*, 27(8):735–741, 2009.
- [10] R. Machné, A. Finney, S. Müller, J. Lu, S. Widder, and C. Flamm. The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks. *Bioinformatics*, 22(11):1406, 2006.
- [11] U. Nowak and P. Deuffhard. Towards parameter identification for large chemical reaction systems. In P. Deuffhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, 1983.
- [12] U. Nowak and P. Deuffhard. Numerical identification of selected rate constants in large chemical reaction systems. *Applied Numerical Mathematics*, 1(1):59–75, 1985.
- [13] U. Nowak and L. Weimann. A family of Newton codes for systems of highly nonlinear equations. ZIB-report 91-10, Zuse Institute Berlin (ZIB), 1991.
- [14] L. Strömbäck and P. Lambrix. Representations of molecular pathways: an evaluation of SBML, PSI MI and BioPAX. *Bioinformatics*, 21(24):4401, 2005.

A Problem Description

A.1 Large Kinetic Networks in Systems Biology

A major topic in systems biology is the study of the dynamical evolution of bio-chemical mechanisms within a well-defined, biology-related context. The bio-chemical mechanisms in such a compound under consideration are typically given as a, possibly huge, set of chemical reactions between numerous species. Thus, the set of chemical reactions are the building blocks of the system model forming a large kinetic network. Assuming the general principle of mass action kinetics, this large network can readily be transformed to a system of n ordinary differential equations (ODEs) leading to the initial value problem (IVP)

$$y' = f(y; p), \quad y(t_0) = y_0 \quad (1)$$

where the right-hand side, f , denotes the dependence of the change in the species vector, y' , on both the species, y , and the parameter vector, p , of dimension q . The initial condition vector, y_0 , has the same dimension as the species vector, y . In BioPARKIN, the ODE systems are solved numerically with LIMEX, a linear implicit Euler method with extrapolation [5].

The first step in Systems Biology often involves developing a suitable model description, f , that will not be dealt with in detail here. Instead, it is assumed that some discrete experimental data (in form of species concentration versus time),

$$(\tau_1, z_1), \dots, (\tau_M, z_M), \quad (2)$$

is available. Note that frequently, only a certain amount of the n species concentration are measurable observables, if at all. The task at hand now reduces to quantify the q unknown components of the parameter vector, p , by comparison between model, synthetic values and given, measured data.

A.2 Multiple Experiments

The design of experiments includes almost always different conditions such that the effects of these different conditions on the system under investigation can be observed and studied. In the simplest case, calibration measurements might be necessary, for example, or data related to different initial conditions, $y_{0,1}, y_{0,2}, \dots, y_{0,\nu}, \dots$, are given. Numerically, these situations can be handled by the concatenation of several IVPs,

$$y'_\nu = f_\nu(y_\nu; p), \quad y_\nu(t_{0,\nu}) = y_{0,\nu}, \quad \nu = 1, 2, \dots, \quad (3)$$

very similar to the management of breakpoints / events (see below). If required, the solution y_ν corresponding to the (virtual) initial timepoint, $t_{0,\nu}$, can readily be shifted to the (original) initial time, t_0 , for comparison or plotting purpose.

A.3 Breakpoint Handling

A sudden event (maybe from outside the biological system) is handled by introducing a breakpoint, $t_b > t_0$, and subsequently, splitting the ODE system into a y^- -part for $t_0 < t < t_b$, and a y^+ -part for $t_b < t$,

$$(y^-)' = f(y^-; p), \quad y^-(t_0) = y_0 \quad (4)$$

$$(y^+)' = f(y^+; p), \quad y^+(t_b) = g(y^-; p) \quad (5)$$

where $g : \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$ is a mapping of the initial conditions; possibly dependent on the parameter vector, p . The approach of splitting the ODE system with respect to time also applies in the case of multiple experiments, at least for the cases described above.

A.4 Sensitivity Matrix

The dependence of the solution, $y(t; p)$, on the parameters, p , is characterised by the sensitivity (n, q) -matrix, $S = S(t)$, defined by

$$S_{ij}(t) = \frac{\partial y_i}{\partial p_j}(t). \quad (6)$$

The matrix can readily be computed as solution to the variational equation

$$S' = f_y(y; p) S + f_p(y; p), \quad S(t_0) = 0 \quad (7)$$

Consequently, the variational equation associated with a transformed system reads

$$\begin{aligned} \tilde{S}' &= \tilde{f}_y(y; u) \tilde{S} + \tilde{f}_u(y; u) \\ &= f_y(y; \varphi(u)) \tilde{S} + f_p(y; \varphi(u)) \varphi'(u), \quad \tilde{S}(t_0) = 0 \end{aligned} \quad (8)$$

showing clearly the influence of the transformation, φ , upon the sensitivity matrix, $\tilde{S}$.

In the case that breakpoints / events are specified, the computation of the sensitivity matrix also splits into the computation of S^- and S^+ ,

$$(S^-)' = f_y(y^-; p) S^- + f_p(y^-; p) \quad (9)$$

$$S^-(t_0) = 0, \quad (10)$$

$$(S^+)' = f_y(y^+; p) S^+ + f_p(y^+; p) \quad (11)$$

$$S^+(t_b) = g_y(y^-; p) S^-(t_b) + g_p(y^-; p) \quad (12)$$

respectively.

Often, model species and model parameters cover a broad range of physical units and their values can vary over orders of magnitude. To achieve comparability, the absolute sensitivities have to be normalized by the absolute values of species and parameters to obtain relative sensitivities:

$$S_{ij}(t) = \left(\frac{\partial y_i}{\partial p_j} \right) (t) \cdot \frac{\max(|p_j|, trsh(p_j))}{\max(\max_{t \in I} |y_i(t)|, trsh(y_i))} \quad (13)$$

In this formula the user specified threshold values for parameters and species occur. I is the integration time interval. It is always the relative sensitivities that are displayed in the *Results Window*.

A.5 Parameter Identification

Assume there are M experimental data points varying in the selected component at different time points,

$$z_i = y_{j_i}(\tau_i), \quad i = 1, \dots, M, \quad j_i \in 1, \dots, N,$$

associated with corresponding measurement tolerances δz_i . Here $y_{j_i}(\tau_i)$ denotes the measurement of component y_{j_i} at time τ_i . The M to N mapping j_i assigns to every measurement time point τ_i one of the N components of y .

Remark: In PARKIN the measurement tolerances are computed as

$$\delta z_i = \max(\eta \cdot \max_i y_{j_i}(\tau_i), y_{j_i}(\tau_i)),$$

with some small safety factor η . In NLSCON the measurement tolerances are computed as

$$\delta z_i = \max(\max_{t \in I} y_{j_i}(t), \text{trsh}(y_{j_i})),$$

with some user specified threshold trsh .

Parameter identification is equivalent to solving the least squares minimization problem

$$\|F(p)\|_2^2 = \sum_{i=1}^M \left(\frac{y_{j_i}(t_i, p) - z_i}{\delta z_i} \right)^2 \rightarrow \min, \quad (14)$$

or, alternatively,

$$I(p) = F(p)^T F(p) \rightarrow \min_p, \quad (15)$$

where $F(p)$ is a vector of length M with entries

$$F_i(p) = \frac{y_{j_i}(t_i) - z_i}{\delta z_i}. \quad (16)$$

That means we want to minimize the relative deviation of model and data at the measurement time points t_i . The above problem, which is usually highly nonlinear in p , can be solved by affine covariant Gauss-Newton iteration [1] where each iteration step k requires the solution of a linear least squares problem,

$$\|J(p^k)\Delta p^k + F(p^k)\|^2 \rightarrow \min, \quad (17)$$

$$p^{k+1} = p^k + \lambda_k \Delta p^k, \quad k = 0, 1, 2, \dots \quad (18)$$

The update step uses a customized QR decomposition for solving the Jacobian system, especially in the rank-deficient case. The i -th row of the Jacobian ($M \times q$)-matrix $J(p)$ has the form

$$J(p)(i, :) = \nabla_p y_{j_i}(t_i; p), \quad (19)$$

thus representing the sensitivity of the solution y with respect to the parameters p at the time points of measurements. An analysis of the matrix $J(p)$ gives some hints whether the current combination of model and data will permit an actual identification of the parameters. Parameters with very small sensitivity have nearly no influence on the solution and can therefore not be estimated. In this case the entries of the corresponding column in $J(p)$ (and thus the weighted l_2 column norm) are almost zero. Furthermore, some of the parameters might be linearly dependent, which leads to nearly identical columns in $J(p)$. In both cases the matrix $J(p)$ will be singular or, from a numerical point of view, nearly singular.

Linearly independent parameters can be identified by analyzing their subcondition. Let us consider the QR -decomposition of $J(p)$. By a suitable permutation of the matrix columns of $J(p)$, the diagonal

elements of the upper triangular matrix R can be ordered in the form $|r_{11}| \geq |r_{22}| \geq \dots \geq |r_{qq}|$. The subcondition of parameter p_l is given by

$$sc_l = |r_{11}|/|r_{ll}|. \quad (20)$$

Thus, the permutation of matrix columns corresponds to a new ordering of parameters according to increasing subcondition. The subcondition indicates whether a parameter can be estimated from the given data or not. Only those parameters can be estimated for which

$$sc_l < 1/\epsilon, \quad (21)$$

where $\epsilon > 0$ is the relative precision of the Jacobian $J(p)$ [3].

Several safeguards and convergence monitors are in place for the gauss-Newton approach:

- Natural monotonicity test via natural level function
- Control of asymptotic convergence rate
- Adaptive rank strategies by controlling the subcondition
- Adaptive trust region strategies for relaxation estimation

These techniques for solving a nonlinear least squares problem are implemented in the codes NLSCON [1] and PARKIN [12, 4]. Numerical experiments in [13] have shown that it is recommended to activate *Automatic Row Scaling of Jacobian* when NLSCON is used.

A.6 Parameter Constraints

In order to enforce constraints such as positivity or upper and lower bounds on the unknown parameters to be determined in the model, a (differentiable) transformation, $\varphi : \mathbb{R}^q \rightarrow \mathbb{R}^q$, can be applied resulting in a different parametrisation, u , of the model ODE system,

$$p = \varphi(u), \quad y' = f(y; \varphi(u)) = \tilde{f}(y; u) \quad (22)$$

A global positivity constraint on the parameter vector, p , can be achieved for example by the (component-wise) exponential transformation

$$p_i = \exp(u_i), \quad i = 1, \dots, q \quad (23)$$

To impose an upper and a lower bound, A and B , respectively, a sinusoidal transformation

$$p_i = A + \frac{B - A}{2} (1 + \sin u_i), \quad i = 1, \dots, q \quad (24)$$

can be used. For a single bound, C , as last example in this section, a root square transformation

$$p_i = C \pm \left(1 - \sqrt{1 + u_i^2}\right), \quad i = 1, \dots, q \quad (25)$$

(with the upper sign for an upper bound and the lower sign for a lower bound) is possible.

Note that the application of any transformation of the parameters obviously change the sensitivities of the parameters to the dynamical evolution of the ODE system, as will be shown next. Therefore, it is strongly recommended that parameter constraints should only be applied in order to prevent the parameter vector components, p_i , from taking on really unphysical values.